REMARKS

Claims 1-15 were examined and are rejected. Without prejudice, and without acknowledging agreement with or acquiescence to the rejections, applicants hereby amend independent claims 1, 7, and 12 and dependent claim 13, and cancel dependent claims 8-11, 14, and 15. All claim changes are supported by applicants' originally-filed application at, for example, page 3, lines 32-34, page 5, lines 4-9 and lines 32-35, page 7, lines 22-32, page 9 starting at line 34 through page 10, line 11, and FIG. 2.

REJECTIONS BASED ON WESSMAN AND DATE

Claims 1-3 and 6-15 are rejected under 35 USC § 103(a) over U.S. Patent No. 7,111,005 to Wessman ("Wessman") in view of Date, "An Introduction to Database Systems." ("Date"). Of these rejected claims, claims 1, 7, and 12 are independent claims, with the rest of the rejected claims depending directly or indirectly from claims 1, 7, or 12.

Applicants request reconsideration of these rejections with regard to amended claims 1, 7, and 12, at least because each of these independent claims recites "forming a restricting character set on the basis of the data type."

Forming the restricting character set is a separate and distinct step involving determining which characters of the data type of the data element are to define the restricting character set (page 5, lines 32-35). That determination can depend on a variety of factors, including the makeup of the characters in the database, the particular attributes of the data elements, or the type of encrypting that is to be performed. See, for example, as-filed application at page 7, line 22-26.

In this way, forming the restricting character set allows the restricting character set to be data and encryption sensitive (see below). Furthermore, forming the restricting character set

permits data elements in a database to be encrypted and stored back into the same data element location in the database without changing the data type of the data element (page 7, lines 11-15). Because the structure of the database remains the same, applications which depend on the database need not be rewritten to integrate with the encrypted data elements and the encryption is therefore transparent to the applications. Id.

Forming a restricting character set can include providing an index value for all characters that are used in a database, for example, all letters, numbers, special characters, control characters, etc. that are used in a database (page 7, line 22-26). Forming the restricting character set could also be done by taking the ASCII value of each character used in a database and adding that value to the restricting character set. Id at 27-32. For example, the ASCII value for "A" can be determined to be "65" in data element "ABC", and then "65" can be added to the restricting character set. Id.

Forming a restricting character set can include defining index values comprised of characters 0-9 and adding them to the restricting character set to represent each character in a data element (page 5, lines 9-13). This is useful for encryption algorithms like DES, which require characters to be converted to numerals (page 8, lines 1-4).

Forming a restricting character set can help ensure that the encrypted values of two single character strings with adjacent characters will not be sequential (page 8, lines 5-9). Here, the index values can be shuffled based on a portion of the encryption key. Id. For example, the data in the restricting character set can be rippled from left to right and then from right to left (page 8, lines 9-13). This can prevent the character "b" from being encrypted as "y" whenever "a" is encrypted as "x".

According to the Office Action, Wessman does not explicitly disclose forming a restricting character set on the basis of the data type of the data element, and encrypting a first character string into a second character string, each character in the second character string being

selected from a restricting character set. See Office Action, page 3, paragraph 1. However, it is asserted in the Office Action that these features are inherent in Wessman. Id.

It may be that databases inherently determine whether incoming characters to be stored in a data column are valid members of the data type of the data column. If the incoming characters are not valid members of the data type, it may be that a database will reject the characters. If the characters are valid, it may be that a database accepts the characters and stores them. The Office Action seems to indicate that rejecting and accepting incoming characters is the same as forming a restricting character set. See Office Action, page 3, paragraph 1. Applicants submit that rejecting and accepting incoming characters is not the same as forming a restricting character set. Forming a restricting character set can be controlled by a variety of factors, including the makeup of the characters in the database, the particular attributes of the data elements, and/or the type of encrypting that is to be performed, as indicated above.

Furthermore, Wessman does not inherently create a second character string to store in the database, each character of the second character string being selected from a restricting character set. Wessman does not teach or suggest selecting characters from a restricting character set when storing characters. Instead, as described above, Wessman merely accepts or rejects incoming characters based on whether or not they are valid members of the data type of the data column.

Thus, Wessman does not disclose forming a restricting character set on the basis of the data type of the data element when encrypting the data elements, and encrypting a first character string into a second character string, each character in the second character string being selected from a restricting character set.

Date does not cure this defect of Wessman. In fact, Date states that "any attempt to introduce an attribute value into the database that is not a value of the relevant data type will simply be rejected." Date, page 23, paragraphs 1-2. Date in no way teaches or suggests forming a restricting character set during encrypting of the data element, and encrypting a first character

string into a second character string, each character in the second character string being selected from a restricting character set.

Given that Wessman and Date both fail to teach or suggest at least this same aspect of the independent claims, no combination of Wessman and Date could have or would have rendered these claims obvious.

Claim 4 is rejected under 35 USC § 103(a) over Wessman in view of Date as applied to claim 1, and in further view of Schneier, "Applied Cryptography" ("Schneier"). Applicants traverse.

Claim 4 depends on claim 1. As discussed above, Wessman and Date do not teach or suggest all the elements of claim 1. Schneier does not cure this defect. Schneier merely describes a one-time pad method in which the encryption is the addition modulo 26 of the message character with a key sequence character. See Schneier, page 15. The key sequence is randomly generated and it is not based on the data type of the data. Id. The receiver reverses the operation to decrypt the message. Id. The encryption is called a one-time pad because the key sequence is destroyed after a single use. Id. Therefore, Wessman, Date, and Schneier, either alone or in combination, do not teach or suggest all the elements of claim 4. Accordingly, a prima facie case of obviousness does not exist.

Claim 5 is rejected under 35 USC § 103(a) over Wessman in view of Date and Schneier as applied to claim 4, and in further view of U.S. Patent No. 4,866,707 to Marshall et al. ("Marshall"). Applicants traverse.

Claim 5 indirectly depends on claim 1 and directly depends on claim 4. As discussed above, Wessman, Date, and Schneier do not teach or suggest all the elements of claim 1 or claim 4. Marshall does not cure this defect. Instead, Marshall describes Cipher Block Chaining, a method for encrypting information using a base key and a message key to create an initialization vector (IV). Marshall, column 9, lines 13-27. The IV is encrypted again under the same base key

to create an encryption key. Id. The message is encrypted using the encryption key. Id. The receiver uses the same base key and message key, sent in clear text, to create the IV and the decryption key, which is used to decrypt the message. Id. Thus, Marshall does not teach or suggest at least forming a restricting character set based on the data type of the data element. Therefore, Wessman, Date, Schneier, and Marshall, either alone or in combination, do not teach or suggest all the elements of claim 5. Accordingly, a prima facie case of obviousness does not exist.

REJECTIONS BASED ON MORAR AND DATE

Claims 1-3 and 7-15 are rejected under 35 USC § 103(a) over U.S. Patent No. 6,678,822 to Morar ("Morar") in view of Date. Of these rejected claims, claims 1, 7, and 12 are independent claims, with the rest of the rejected claims depending directly or indirectly from claims 1, 7, or 12.

Applicants request reconsideration of these rejections with regard to amended claims 1, 7, and 12, at least because these claims recite encrypting (or reencrypting) first character string in a particular column of a database into an encrypted second character string, and storing the second character string at the particular column of the first character string in the database. In this way, a database can be secured without changing the structure of the database because the data can be encrypted and stored back into the same location in the database (page 7, lines 11-15; see above describing using the same data type for the encrypted data). Because the structure is unchanged, there is no need to reintegrate and reprogram applications that depend on the database. Id.

Morar does not teach or suggest encrypting a first character string in a particular column of a database into an encrypted second character string, and storing the second character string at the particular column of the first character string in the database. Instead, Morar discloses a first data processing system located within a trusted environment and a second data processing system

located in an untrusted environment. See Morar, column 2, lines 37-43. An agent operating in the first data processing system locates an information container in the trusted environment and identifies predefined restricted information in the information container. Id. at 43-51. A copy of the information container is created in which all information identified as restricted is obscured in the new information container. Id. at 51-54. The new information container with the obscured restricted information is made available to the untrusted environment for use. Id. at 54-57.

Applicants submit that creating a new version of an information container in a trusted environment, wherein predefined restricted information is obscured in the new information container and used in an untrusted environment is different than reading data at a particular column from a database, encrypting (or reencrypting) the data, and storing the encrypted data back into the particular column of the database. Morar makes a copy of the information container, and obscures restricted information in the **copy** of the information container, whereas the claims recite encrypting the data at the same location in the same database. In fact, because Morar obscures a copy, it simply could not be used to secure a database in a way that prevents the need to reintegrate and reprogram applications that depend on the database.

Put another way, in Morar, the **original data is not changed**, whereas the claims recite converting the original data into encrypted data and restoring the data at the same location in the database. Morar is also different because the data in the trusted environment is unsecured, whereas in the claims, the data could have previously been encrypted and secured.

Date does not cure this defect of Morar. Date may describe encryption in broad terms and it may generally discuss databases, but Date in no way teaches or suggests encrypting a first character string in a particular column of a database into an encrypted second character string, and storing the second character string at the particular column of the first character string in the database.

Given that Morar and Date both fail to teach or suggest at least this same aspect of the

independent claims, no combination of Morar and Date could have or would have rendered these claims obvious.

Claim 4 is rejected under 35 USC § 103(a) over Morar in view of Date as applied to claim 1, and in further view of Schneier. Applicants traverse.

Claim 4 depends on claim 1. As discussed above, Morar and Date do not teach or suggest all the elements of claim 1. Schneier does not cure this defect. Schneier may describe encryption in broad terms and it may generally discuss databases, but Schneier in no way teaches or suggests encrypting a first character string in a particular column of a database into an encrypted second character string, and storing the second character string at the particular column of the first character string in the database. Therefore, Morar, Date, and Schneier, either alone or in combination, do not teach or suggest all the elements of claim 4. Accordingly, a prima facie case of obviousness does not exist.

Claim 6 is rejected under 35 USC § 103(a) over Morar in view of Date as applied to claim 1, and in further view of Schneier. Applicants traverse.

Claim 6 depends on claim 1. As discussed above, Morar and Date do not teach or suggest all the elements of claim 1. As described above, Schneier does not cure this defect. Therefore, Morar, Date, and Schneier, either alone or in combination, do not disclose all the elements of claim 6. Accordingly, a prima facie case of obviousness does not exist.

Claim 5 is rejected under 35 USC § 103(a) over Morar in view of Date and Schneier as applied to claim 4, and in further view of Marshall. Applicants traverse.

Claim 5 indirectly depends on claim 1 and directly depends on claim 4. As discussed above, Morar, Date, and Schneier do not teach or suggest all the elements of claim 1 or claim 4. Marshall describes preventing accidental or deliberate interference of messages in communication networks (see, for example, Marshall, column 1, lines 22-53). Marshall in no way teaches or suggests encrypting a first character string in a particular column of a database

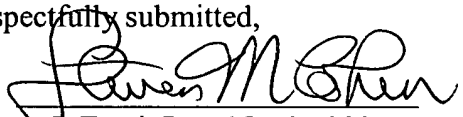
into an encrypted second character string, and storing the second character string at the particular column of the first character string in the database. Therefore, Morar, Date, Schneier, and Marshall, either alone or in combination, do not teach or suggest all the elements of claim 5. Accordingly, a prima facie case of obviousness does not exist.

CONCLUSION

In view of the foregoing, applicants submit that claims 1-7, 12, and 13 are in condition for allowance and should be allowed in due course. Applicants submit that no extension petition or fees are due to have this Response to Final Action entered and considered. However, if for any reason an extension petition is required and/or a fee is required to have this response entered and considered, please consider this a conditional petition for the necessary extension and/or conditional authorization to charge Deposit Account No. 04-1105 for any required fee.

Dated: January 29, 2007

Respectfully submitted,

By: 

Robert J. Tosti, Reg. No. 35,393
Steven M. Cohen, Reg. No. 59,503
Edwards Angell Palmer & Dodge LLP
P.O. Box 55874
Boston, MA 02205
Direct line: (617) 517-5584

Customer No. 21,874